МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №4-7

по дисциплине

‘Системы искусственного интеллекта‘

Выполнил:
Студент группы P33312
Соболев Иван
Александрович
Преподаватель:
Кугаевских Александр
Владимирович

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

Модуль 2.

Лабораторная работа 1. Метод линейной регрессии

Введение

- Получите и визуализируйте статистику по датасету (включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили).
- Проведите предварительную обработку данных, включая обработку отсутствующих значений, кодирование категориальных признаков и нормировка.
- Разделите данные на обучающий и тестовый наборы данных.
- Реализуйте линейную регрессию с использованием метода наименьших квадратов без использования сторонних библиотек, кроме NumPy и Pandas (для использования коэффициентов использовать библиотеки тоже нельзя). Использовать минимизацию суммы квадратов разностей между фактическими и предсказанными значениями для нахождения оптимальных коэффициентов.
- Постройте **три модели** с различными наборами признаков.
- Для каждой модели проведите оценку производительности, используя метрику коэффициент детерминации, чтобы измерить, насколько хорошо модель соответствует данным.
- Сравните результаты трех моделей и сделайте выводы о том, какие признаки работают лучше всего для каждой модели.

Описание метода

Метод линейной регрессии - это статистический метод, используемый для определения связи между зависимой и независимыми переменными. Принцип работы метода заключается в построении линии наилучшего соответствия данных.

Псевдокод метода

1. Подготовить данные: разделить данные на тренировочный и тестовый наборы.
2. Выбрать модель: определить вид модели линейной регрессии.
3. Обучить модель: подобрать параметры модели с использованием тренировочных данных.
4. Оценить модель: оценить точность модели на тестовом наборе данных.
5. Применить модель: использовать обученную модель для прогноза значений зависимой переменной.

Результаты выполнения

In [13]:
```
# Модель по всем признакам
y_pred, r2,sum_of_squares = perform_linear_regression(None,X_train, X_test, y_train, y_test)
print('Коэффициент детерминации:', r2)
print('Предсказания:', y_pred)
print('Сумма квадратов',sum_of_squares)
```

```
Коэффициент детерминации: 0.9889053868454428
Предсказания: [86.43804874 94.91719465 56.41053439 ... 64.34202553 46.46900286
 65.97643882]
Сумма квадратов 8194.94125307
```

In [14]:
```
columns = 'Previous Scores'
# Модель по Previous Scores
y_pred, r2,sum_of_squares = perform_linear_regression(columns,X_train, X_test, y_train, y_test)
print('Коэффициент детерминации:', r2)
print('Предсказания:', y_pred)
print('Сумма квадратов',sum_of_squares)
```

```
Коэффициент детерминации: 0.8382697969226347
Предсказания: [78.11032906 84.1924904  58.85015149 ... 77.0966355  57.83645794
 76.08294195]
Сумма квадратов 119460.6332462965
```

In [16]:
```
columns = 'Hours Studied,Previous Scores'
# Модель по Previous Scores и Hours Studied
y_pred, r2,sum_of_squares = perform_linear_regression(columns,X_train, X_test, y_train, y_test)
print('Коэффициент детерминации:', r2)
print('Предсказания:', y_pred)
print('Сумма квадратов',sum_of_squares)
```

```
Коэффициент детерминации: 0.9863248845979447
Предсказания: [86.80880077 95.7835485  56.00808339 ... 65.79140967 46.41797023
 67.62857445]
Сумма квадратов 10101.00719940508
```

In [17]:
```
columns = 'Hours Studied,Previous Scores,Motivation'
# Модель по Previous Scores, Hours Studied и Motivation
y_pred, r2,sum_of_squares = perform_linear_regression(columns,X_train, X_test, y_train, y_test)
print('Коэффициент детерминации:', r2)
print('Предсказания:', y_pred)
print('Сумма квадратов',sum_of_squares)
```

```
Коэффициент детерминации: 0.9869521442968681
Предсказания: [86.77300848 95.74173466 56.52246844 ... 65.08052036 46.76887153
 67.6176465 ]
Сумма квадратов 9637.687179906736
```

Выводы

Заметим, что коэффициент детерминации сильно повысился за счёт Previous Scores и Hours Studied, можем сделать вывод, что успеваемость зависит от имеющихся знаний студента(предыдущих оценок), и также от того, сколько часов он потратил на учебу. Если у студента хорошие знания и мало учился, то скорее успеваемость у него будет немного меньше, а если хорошо подготовился, то будет примерно такой же балл. Также мотивация немного поднимает коэффициент детерминации, следовательно, чем выше мотивация, тем выше будет успеваемость.

Примеры использования метода

Метод линейной регрессии может быть полезен во многих ситуациях. Например, он может быть использован для прогнозирования продаж, основываясь на данных о рекламных затратах или других факторах, таких как время года, погода и т. д. Также, данный метод может быть применен для анализа взаимосвязей между различными переменными, таких как цена товара, его характеристики и спрос на рынке. Выбор метода линейной регрессии обусловлен его простотой и понятностью, а также хорошей прогностической способностью.

Лабораторная работа 2. Метод k-ближайших соседей (k-NN)

Введение

- Проведите предварительную обработку данных, включая обработку отсутствующих значений, кодирование категориальных признаков и масштабирование.
- Реализуйте метод k-ближайших соседей без использования сторонних библиотек, кроме NumPy и Pandas.
- Постройте две модели k-NN с различными наборами признаков:
  - Модель 1: Признаки случайно отбираются .
  - Модель 2: Фиксированный набор признаков, который выбирается заранее.
- Для каждой модели проведите оценку на тестовом наборе данных при разных значениях k. Выберите несколько различных значений k, например, k=3, k=5, k=10, и т. д. Постройте матрицу ошибок.

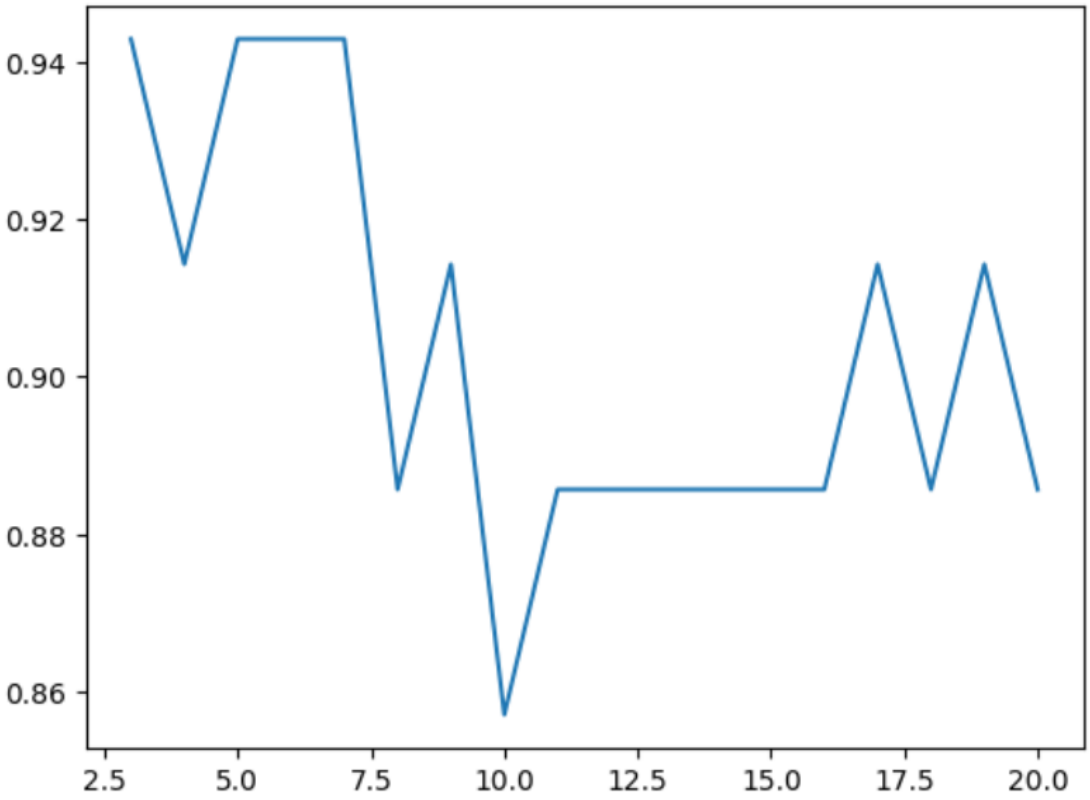
Описание метода

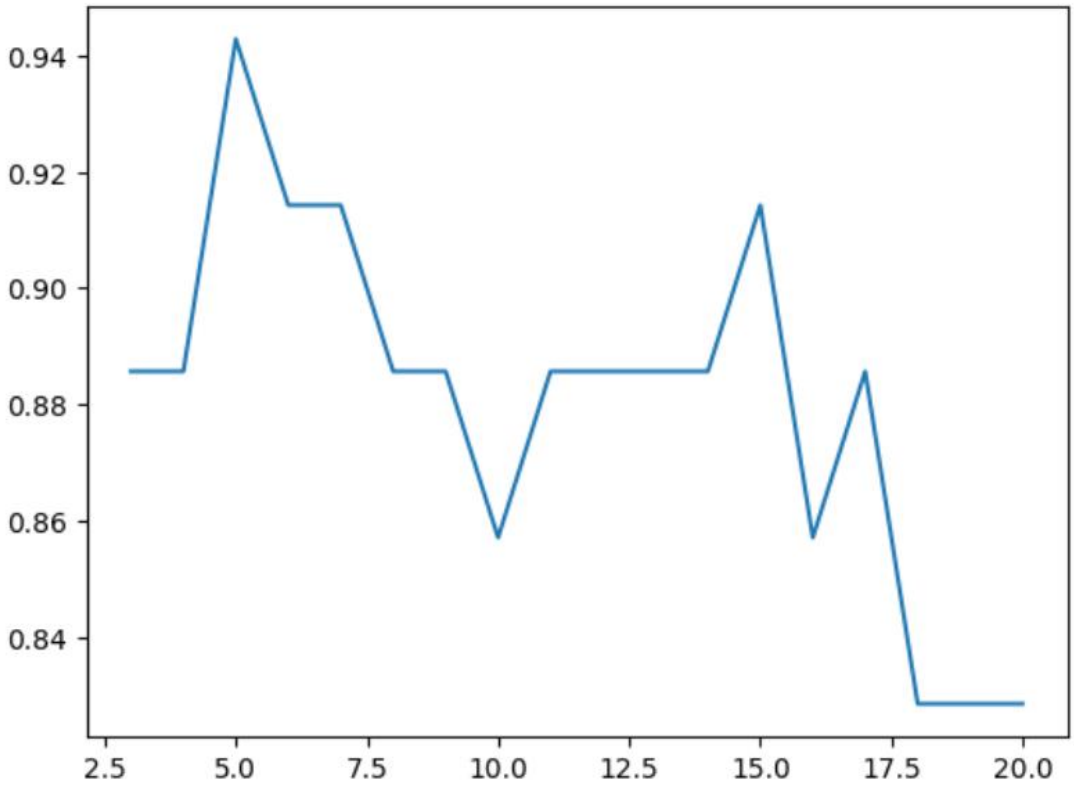
Метод k-ближайших соседей используется для классификации объектов на основе их близости к примерам обучающей выборки. Основной принцип работы метода заключается в нахождении k ближайших соседей объекта и отнесении его к классу, который наиболее часто встречается среди этих соседей.

Псевдокод метода

1. Для каждого объекта из обучающей выборки:
2. Вычислить расстояние между объектом из обучающей выборки и новым объектом.
3. Отсортировать объекты из обучающей выборки по возрастанию расстояния.
4. Выбрать k ближайших соседей.
5. Результирующее значение целевой переменной для нового объекта будет равно значению целевой переменной, которое имеют наиболее часто встречающиеся среди его k ближайших соседей.

Результаты выполнения





ВЫВОД

Можно заметить, что с увеличением количества ближайших соседей показатель f1_score падает. Оптимальным количеством соседей является 5-7 для данного набора тренировочных и тестовых данных.

Примеры использования метода

Метод k-ближайших соседей может быть полезен в следующих ситуациях:

Когда у нас есть обучающая выборка, для которой известны значения целевой переменной, и мы хотим классифицировать новый объект.

Когда данные имеют сложную структуру и требуют нелинейной модели для классификации или регрессии.

Лабораторная работа 3. Деревья решений

Введение

1. Для студентов с четным порядковым номером в группе – датасет с классификацией грибов, а нечетным – датасет с данными про оценки студентов инженерного и педагогического факультетов (для данного датасета нужно ввести метрику: студент успешный/неуспешный на основании грейда)
2. Отобрать **случайным** образом sqrt(n) признаков
3. Реализовать без использования сторонних библиотек построение дерева решений (numpy и pandas использовать можно, использовать списки для реализации дерева - нельзя)
4. Провести оценку реализованного алгоритма с использованием Accuracy, precision и recall
5. Построить AUC-ROC и AUC-PR (в пунктах 4 и 5 использовать библиотеки нельзя)

Описание метода

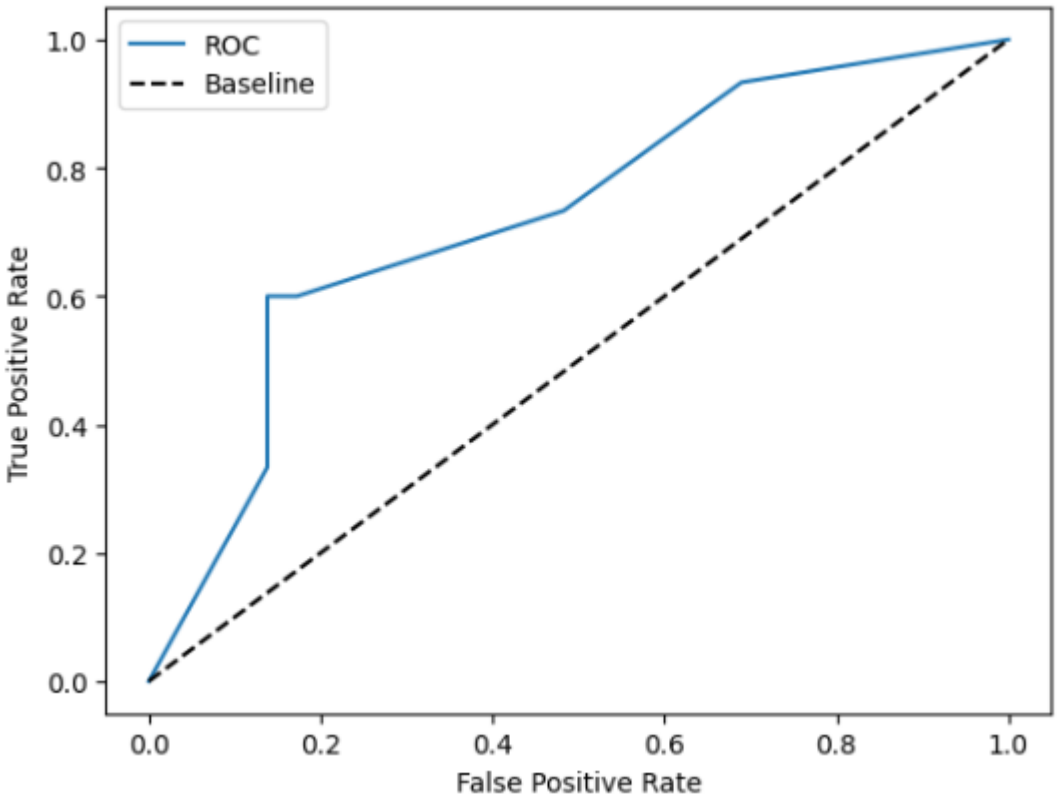
Метод деревьев решений является методом машинного обучения, который основывается на создании дерева, в котором каждый узел представляет условие или атрибут, а каждое ребро - результат этого условия. Дерево решений используется для прогнозирования или принятия решений на основе заданных данных. Он может применяться как для задач классификации, так и для задач регрессии.

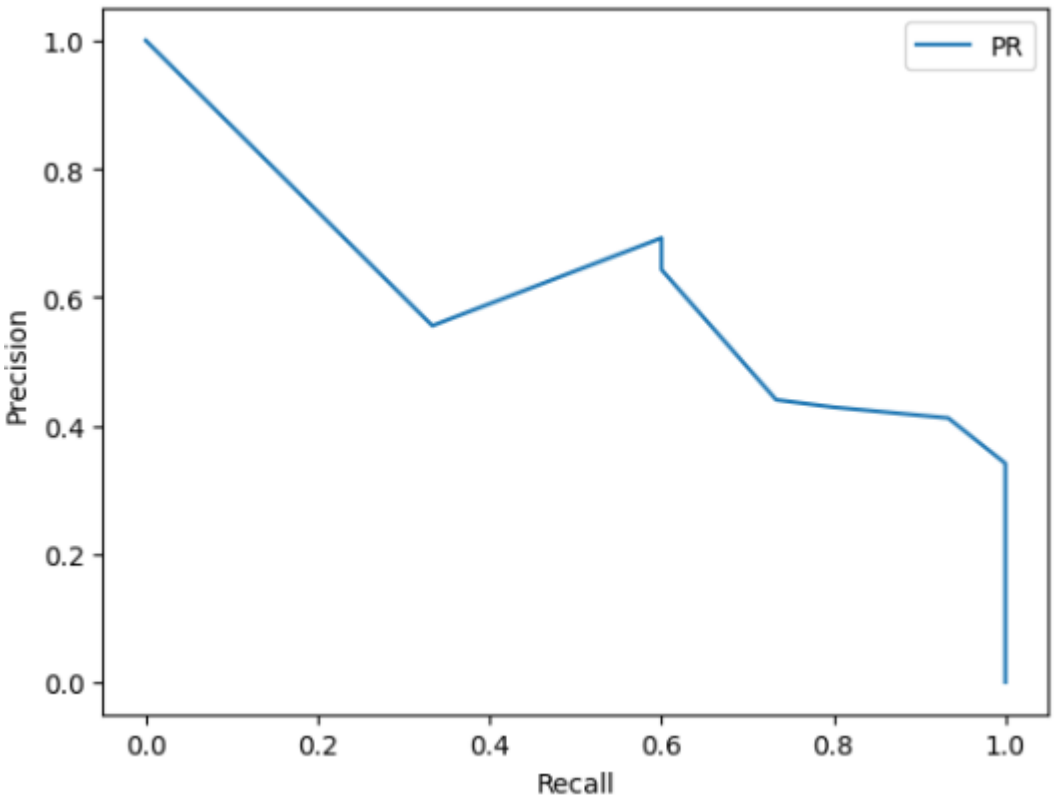
Псевдокод метода

[A] Если все объекты в выборке относятся к одному классу, вернуть узел с этим классом
[B] Если все атрибуты уже рассмотрены, вернуть узел с наиболее часто встречающимся классом
[C] Иначе
  [D] Найти атрибут с наибольшим приростом информации
  [E] Создать узел для выбранного атрибута
  [F] Для каждого значения атрибута создать потомок на дереве
  [G] Рекурсивно применить алгоритм для новых потомков

Результаты выполнения

```
[[25.  6.]
 [ 4.  9.]]
Accuracy: 0.7727272727272727
Precision 0.6923076923076923
Recall: 0.6
```





Примеры использования метода

Метод деревьев решений может быть полезен в следующих ситуациях:

Классификация клиентов по их покупательскому поведению для оптимизации маркетинговых стратегий.

Прогнозирование вероятности оттока клиентов на основе их активности в приложении.

Определение причин аварий на основе данных о состоянии системы.

Решение задачи обнаружения мошеннических операций на основе истории транзакций.

Лабораторная работа 4. Логистическая регрессия

Введение

- Разделите данные на обучающий и тестовый наборы в соотношении, которое вы считаете подходящим.
- Реализуйте логистическую регрессию "с нуля" без использования сторонних библиотек, кроме NumPy и Pandas. Ваша реализация логистической регрессии должна включать в себя:
  - Функцию для вычисления гипотезы (sigmoid function).
  - Функцию для вычисления функции потерь (log loss).
  - Метод обучения, который включает в себя градиентный спуск.
  - Возможность варьировать гиперпараметры, такие как коэффициент обучения (learning rate) и количество итераций.

1. Исследование гиперпараметров:
   - Проведите исследование влияния гиперпараметров на производительность модели. Варьируйте следующие гиперпараметры:
     - Коэффициент обучения (learning rate).
     - Количество итераций обучения.
     - Метод оптимизации (например, градиентный спуск или оптимизация Ньютона).
2. Оценка модели:
   - Для каждой комбинации гиперпараметров оцените производительность модели на тестовом наборе данных, используя метрики, такие как accuracy, precision, recall и F1-Score.

Описание метода

Логистическая регрессия - это метод машинного обучения, который используется для предсказания вероятности бинарного исхода на основе набора входных переменных. Он основан на логистической функции, которая преобразует линейную комбинацию входных переменных в вероятность отнесения к одному из классов.

Псевдокод метода

Инициализация:
- Инициализировать веса w случайными значениями
- Инициализировать смещение (bias) b нулевым значением
- Установить learning_rate (скорость обучения)

- Установить количество итераций для обучения

Метод sigmoid:
- Принимает на вход массив x
- Возвращает 1 / (1 + exp(-x))

Метод predict:
- Принимает на вход данные x
- Вычисляет значения y_pred путем применения функции sigmoid к (x * w + b)
- Возвращает y_pred

Метод train:
- Принимает на вход данные X и соответствующие метки y
- Для каждой итерации обучения:
  - Вычисляет значения y_pred с помощью метода predict для текущих весов w и смещения b
  - Вычисляет градиенты для каждого веса и смещения с помощью формул градиентного спуска
- Обновляет веса w и смещение b, умножив градиенты на learning_rate
- Возвращает обученные веса w и смещение b

Результаты выполнения

```
        Best parameters:
Learning rate: 0.5
Method: train_with_gradient_descent
Iterations: 1000
Accuracy: 0.8366013071895425
```

ВЫВОДЫ

В ходе работы я реализовал метод логистической регрессии средствами языка Python и библиотек NumPy и Pandas. Были построены модели с различными параметрами. Выбрана модель с наилучшими показателями производительности. При варьировании гиперпараметров возникали различные ситуации, соответственно, различные метрики выходили. Однако, стоит заметить, что метод градиентного спуска давал различные результаты в течение работы. В отличие от него, метод оптимизации Ньютона всегда давал приблизительно равные значения, был более стаблильным. Accuracy реализованной модели превысила accuracy sklearn модели.

Примеры использования метода

Метод логистической регрессии может быть полезен в различных ситуациях, например:

Классификация электронных писем как спам или не спам

Классификация покупателей как потенциальных или не потенциальных клиентов

Определение вероятности возникновения заболевания на основе набора медицинских параметров

Сравнение методов

Сравнительный анализ методов

Линейная регрессия:

- Преимущества: Простота реализации и интерпретации, хорошая производительность на данных с линейной зависимостью, подходит для предсказания непрерывных значений.

- Ограничения: Линейная регрессия не справляется с нелинейными зависимостями, чувствительна к выбросам и шуму в данных.

Логистическая регрессия:

- Преимущества: Хорошо работает при бинарной классификации, расчет вероятности принадлежности к классу, относительно проста для интерпретации.

- Ограничения: Плохо работает с данными, имеющими сложную нелинейную структуру, требует линейной разделимости классов.

Деревья решений:

- Преимущества: Может работать с любыми типами данных, обработка пропущенных значений и выбросов, хорошо интерпретируемый результат, легко обработать категориальные переменные.

- Ограничения: Склонны к переобучению на сложных данных, неустойчивость к небольшим изменениям в данных.

Метод k ближайших соседей:

- Преимущества: Простота реализации, хорошо работает на данных с нелинейной зависимостью, способен обрабатывать выбросы и шум в данных.

- Ограничения: Требуется хранение всего обучающего набора данных, неэффективен при работе с большими объемами данных, требуется определение и настройка значения k.

Примеры лучшего использования каждого метода

- Линейная регрессия может быть эффективна при предсказании цен на недвижимость, где зависимость между факторами и ценой может быть линейной.

- Логистическая регрессия может быть полезна для прогнозирования вероятности оттока клиентов в банковской отрасли или предсказания вероятности заболевания на основе медицинских данных.

- Деревья решений могут быть эффективны при принятии решений о предоставлении кредита, где нужно учитывать множество факторов.

- Метод k ближайших соседей может использоваться для классификации текстовых документов или обработки изображений с нелинейной структурой.

Заключение

В зависимости от типа данных, сложности задачи и требований к интерпретируемости, каждый из этих методов имеет свои преимущества и ограничения. Важно выбирать метод, который наиболее подходит для конкретной задачи и обучать модель с использованием оптимальных гиперпараметров.

Приложения

Код реализованных методов: